Lecture #3

Fields, Meshes, and Interpolation (Part 2)

Hank Childs, University of Oregon
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Project #1

- **Goal:** write a specific image
- **Due:** “today”
- **% of grade:** 2%

How is project #1 going?
Project #2

• On Canvas
• Initially:
  – Wanted to assign today
  – Due Friday
• But: Monday is MLK Day
• And: may not get through enough lecture to assign it.
• So: look at Piazza for update
Office Hours

• Brent still available today … email for an appointment
Classroom change

• We have been offered another classroom...
Looking for notetaker

Hi Everyone,

The AEC is looking for a notetaker for this course. See below.

Thanks!
-Hank

The Accessible Education Center is requesting a peer notetaker for this course. You can earn $25 per credit hour for uploading the notes that you're already taking. If you take clear and comprehensive notes, please go to aec.uoregon.edu to sign up to be a notetaker. You may also enter the CRNs of other classes you are taking to see if there are additional notetaking opportunities.
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Elements of a Visualization

Legend

Display of Data

Reference Cues

Provenance Information
Elements of a Visualization

What is the value at this location? How do you know?

What data went into making this picture?
Scalar Fields

• Defined: associate a scalar with every point in space.
• What is a scalar?
  – A: a real number
• Examples:
  – Temperature
  – Density
  – Pressure

The temperature at 41.2324° N, 98.4160° W is 66°F.

Fields are defined at every location in a space (example space: USA)
Vector Fields

- Defined: associate a vector with every point in space.

- What is a vector?
  - A: a direction and a magnitude

- Examples:
  - Velocity

Typically, 2D spaces have 2 components in their vector field, and 3D spaces have 3 components in their vector field.
An example mesh
An example mesh

Where is the data on this mesh?
(for today, it is at the vertices of the triangles)
An example mesh

Why do you think the triangles change size?
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Anatomy of a computational mesh

- Meshes contain:
  - Cells
  - Points
- This mesh contains 3 cells and 13 vertices
- Pseudonyms:
  - Cell == Element == Zone
  - Point == Vertex == Node
Types of Meshes

Curvilinear

Adaptive Mesh Refinement

Unstructured

We will discuss all of these mesh types more later in the course.
Rectilinear meshes

- Rectilinear meshes are easy and compact to specify:
  - Locations of X positions
  - Locations of Y positions
  - 3D: locations of Z positions

- Then: mesh vertices are at the cross product.

- Example:
  - X={0,1,2,3}
  - Y={2,3,5,6}
Rectilinear meshes aren’t just the easiest to deal with ... they are also very common
Quiz Time

• A 3D rectilinear mesh has:
  – X = {1, 3, 5, 7, 9}
  – Y = {2, 3, 5, 7, 11, 13, 17}
  – Z = {1, 2, 3, 5, 8, 13, 21, 34, 55}

• How many points?  = 5*7*9 = 315
• How many cells?   = 4*6*8 = 192
Definition: dimensions

• A 3D rectilinear mesh has:
  – \( X = \{1, 3, 5, 7, 9\} \)
  – \( Y = \{2, 3, 5, 7, 11, 13, 17\} \)
  – \( Z = \{1, 2, 3, 5, 8, 13, 21, 34, 55\} \)

• Then its **dimensions** are 5x7x9
How to Index Points

• Motivation: many algorithms need to iterate over points.

```c
for (int i = 0 ; i < numPoints ; i++)
{
    double *pt = GetPoint(i);
    AnalyzePoint(pt);
}
```
Schemes for indexing points

<table>
<thead>
<tr>
<th>Logical point indices</th>
<th>Point indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 1.5 2.5 3.5 4.5 5.5</td>
<td>30 31 32 33 34 35</td>
</tr>
<tr>
<td>0.4 1.4 2.4 3.4 4.4 5.4</td>
<td>24 25 26 27 28 29</td>
</tr>
<tr>
<td>0.3 1.3 2.3 3.3 4.3 5.3</td>
<td>18 19 20 21 22 23</td>
</tr>
<tr>
<td>0.2 1.2 2.2 3.2 4.2 5.2</td>
<td>12 13 14 15 16 17</td>
</tr>
<tr>
<td>0.1 1.1 2.1 3.1 4.1 5.1</td>
<td>6  7  8  9 10 11</td>
</tr>
<tr>
<td>0.0 1.0 2.0 3.0 4.0 5.0</td>
<td>0  1  2  3  4  5</td>
</tr>
</tbody>
</table>

What would these indices be good for?
How to Index Points

• Problem description: define a bijective function, F, between two sets:
  – Set 1: \{(i,j,k): 0 \leq i < nX, 0 \leq j < nY, 0 \leq k < nZ\}
  – Set 2: \{0, 1, ..., nPoints-1\}

• Set 1 is called “logical indices”
• Set 2 is called “point indices”

Bijection: for every element in set 1, there is an element in set 2. And vice-versa.

Note: for the rest of this presentation, we will focus on 2D rectilinear meshes.
How to Index Points

• Many possible conventions for indexing points and cells.

• Most common variants:
  – X-axis varies most quickly
  – X-axis varies most slowly
Bijective function for rectilinear meshes for this course

```c
int GetPoint(int i, int j, int nX, int nY)
{
    return j*nX + i;
}
```

Bijective function for rectilinear meshes for this course

```c
int *GetLogicalPointIndex(int point, int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv; // terrible code!!
}
```
int *GetLogicalPointIndex(int point,
    int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv;
}
Quiz Time #2

- A mesh has dimensions 6x8.
- What is the point index for (3,7)? = 45
- What are the logical indices for point 37? = (1,6)

```c
int GetPoint(int i, int j, int nX, int nY)
{
    return j*nX + i;
}

int *GetLogicalPointIndex(int point, int nX, int nY)
{
    int rv[2];
    rv[0] = point % nX;
    rv[1] = (point/nX);
    return rv; // terrible code!!
}
```
• A vector field is defined on a mesh with dimensions 100x100
• The vector field is defined with double precision data.
• How many bytes to store this data?

\[= 100 \times 100 \times 2 \times 8 = 160,000\]
Bijective function for rectilinear meshes for this course

```c
int GetCell(int i, int j, int nX, int nY)
{
    return j*(nX-1) + i;
}
```
Bijective function for rectilinear meshes for this course

```c
int *GetLogicalCellIndex(int cell,
                          int nX, int nY)
{
    int rv[2];
    rv[0] = cell % (nX-1);
    rv[1] = (cell/(nX-1));
    return rv; // terrible code!!
}
```
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Goal: have data at some points & want to interpolate data to any location
Linear Interpolation for Scalar Field $F$

![Diagram showing linear interpolation with points A, X, and B, and function values $F(A)$, $F(X)$, and $F(B)$.]
Linear Interpolation for Scalar Field \( F \)

- General equation to interpolate:
  \[ F(X) = F(A) + t*(F(B)-F(A)) \]
- \( t \) is proportion of \( X \) between \( A \) and \( B \)
  \[ t = (X-A)/(B-A) \]
Quiz Time #4

• $F(3) = 5$, $F(6) = 11$
• What is $F(4)$? $= 5 + (4-3)/(6-3)*(11-5) = 7$

• General equation to interpolate:
  $F(X) = F(A) + t*(F(B)-F(A))$
• $t$ is proportion of $X$ between $A$ and $B$
  $t = (X-A)/(B-A)$
Bilinear interpolation for Scalar Field $F$

- $F(0,0) = 10$
- $F(0,1) = 1$
- $F(1,0) = 5$
- $F(1,1) = 6$
- $F(0.3, 0) = 8.5$
- $F(0.3, 1) = 2.5$
- What is value of $F(0.3, 0.4)$?

$= 6.1$

- General equation to interpolate:
  
  $$F(X) = F(A) + t*(F(B)-F(A))$$

Idea: we know how to interpolate along lines. Let's keep doing that and work our way to the middle.
Outline

- Projects & OH
- Review
- The Data We Will Study (pt 2)
  - Meshes
  - Interpolation
- Cell Location
- Project 2
Cell location

• Problem definition: you have a physical location \((P)\). You want to identify which cell contains \(P\).

• Solution: multiple approaches that incorporate spatial data structures.
  – Best data structure depends on nature of input data.
    • More on this later in the quarter.
Cell location for project 2

• Traverse X and Y arrays and find the logical cell index.
  – X={0, 0.05, 0.1, 0.15, 0.2, 0.25}
  – Y={0, 0.05, 0.1, 0.15, 0.2, 0.25}

• (Quiz) what cell contains (0.17,0.08)?
  = (3,1)
Facts about cell (3,1)

- It’s cell index is 8.
- It contains points (3,1), (4,1), (3,2), and (4,2).
- Facts about point (3,1):
  - It’s location is (X[3], Y[1])
  - It’s point index is 9.
  - It’s scalar value is F(9).
- Similar facts for other points.
- → we have enough info to do bilinear interpolation
Outline

• Projects & OH
• Review
• The Data We Will Study (pt 2)
  – Meshes
  – Interpolation
• Cell Location
• Project 2
Project 2: Field evaluation

• Goal: for point P, find F(P)

• Strategy in a nut shell:
  – Find cell C that contains P
  – Find C’s 4 vertices, V0, V1, V2, and V3
  – Find F(V0), F(V1), F(V2), and F(V3)
  – Find locations of V0, V1, V2, and V3
  – Perform bilinear interpolation to location P
Project 2

• Assigned today, prompt online
• Due January 17th, midnight (→ January 18th, 6am)
• Worth 7% of your grade
• I provide:
  – Code skeleton online
  – Correct answers provided
• You send me:
  – source code
  – output from running your program
What’s in the code skeleton

• Implementations for:
  – GetNumberOfPoints
  – GetNumberOfCells
  – GetPointIndex
  – GetCellIndex
  – GetLogicalPointIndex
  – GetLogicalCellIndex

  – “main”: set up mesh, call functions, create output
What’s not in the code skeleton

```c
// pt: a two-dimensional location
// dims: an array of size two.
// The first number is the size of the array in argument X,
// the second the size of Y.
// X: an array (size is specified by dims).
// This contains the X locations of a rectilinear mesh.
// Y: an array (size is specified by dims).
// This contains the Y locations of a rectilinear mesh.
// F: a scalar field defined on the mesh. Its size is dims[0]*dims[1].
float EvaluationFunction(const float *pt, const int *dims,
                        const float *X, const float *Y, const float *F)
{
    return 0; // IMPLEMENT ME!!
}
```

... and a few other functions you need to implement
Cell-centered data